



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/350,492	07/09/1999	VENKATESH KRISHNAN	10981455-1	8205
22879	7590	01/31/2003	EXAMINER	
HEWLETT PACKARD COMPANY P O BOX 272400, 3404 E. HARMONY ROAD INTELLECTUAL PROPERTY ADMINISTRATION FORT COLLINS, CO 80527-2400			TANG, KENNETH	
		ART UNIT	PAPER NUMBER	
		2127		
DATE MAILED: 01/31/2003				

Please find below and/or attached an Office communication concerning this application or proceeding.

Offic Action Summary	Application N .	Applicant(s)
	09/350,492	KRISHNAN ET AL.
	Examiner Kenneth Tang	Art Unit 2127

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 14 November 2002.
- 2a) This action is FINAL. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-29 is/are pending in the application.
 - 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-29 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on 09 July 1999 is/are: a) accepted or b) objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- 11) The proposed drawing correction filed on _____ is: a) approved b) disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.
- 12) The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

- 13) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.
- 14) Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).
 - a) The translation of the foreign language provisional application has been received.
- 15) Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413) Paper No(s). _____ .
2) <input checked="" type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)
3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .	6) <input type="checkbox"/> Other: _____ .

DETAILED ACTION

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

1. Claims 1, 2 and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nelson (US 2001/0049686) in view of Hinsley (5,930,511).

Referring to claims 1 and 16, Nelson discloses a software system (software system, [0008]) comprising:

- virtual machine (“Java Virtual Machine”, [0039]);
- native threads interface layer which adapts the methods of the threads interface layer to a platform which underlies the software system (Java Native Interface layer, [0039], Fig. 6, 608, and threads, [0039]).

Nelson fails to explicitly teach:

- thread interface layer which has a set of methods that provide thread support according to a standard threads interface

However, it would have been obvious to one of ordinary skill in the art that a “standard threads interface” would be needed for “threads support” between the two tiers (threads interface layer

and native threads interface layer) because coordination and communication between the two layers are necessary for it to work properly as a whole system.

Nelson also fails to explicitly teach:

- Parallel execution of a plurality of software tasks which are adapted to the virtual machine.

However, Hinsley teaches “parallel processing” with “execution of a task” for a “virtual processor” (see Abstract). It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of parallel execution of tasks to a virtual machine for the reason of increasing the speed of the system.

Referring to claim 2, Nelson teaches having a Java-based standard threads interface (Java class, [0037], and Java, Java Native Interface, threads [0039]);

2. Claims 3-6, 11, 17-20, and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nelson (US 2001/0049686) in view of Hinsley (US 5,930,511), and further in view of Delagi (US 3,858,182).

Referring to claims 3 and 17, Nelson fails to explicitly teach maintaining a set of context information for each software task in terms of the virtual machine. However, Delagi teaches saving context information from a previous virtual machine to a current virtual machine (context information, virtual machine, restore previous context, col 8 lines 6-18). It would have been

obvious to one of ordinary skill in the art at the time the invention was made to include the Delagi feature of maintaining context information in terms of the virtual machine to the existing system so that the virtual machine can resume the execution of any program when needed, therefore, increasing efficiency and reducing overhad involved with resuming execution (col 8, lines 6-15).

Referring to claims 4 and 18, Nelson fails to explicitly teach having the context information including a value for each of a set of virtual machine registers associated with a corresponding software task. However, Delagi teaches a system that maps virtual registers by loading appropriate values into processor registers. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include this feature with having each register associated with a corresponding thread to the existing system for the reason of having an organized way for assignment and being easier to keep track of the threads.

3. Claims 5 and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nelson (US 2001/0049686) in view of Hinsley (US 5,930,511), and further in view of Travis (US 5,619,710).

Referring to claims 5 and 19, Nelson fails to explicitly teach the “native threads interface layer maintaining a set of context information for each software task in terms of the platform.” However, Travis teaches using the context information to select a platform, thus being platform dependent (context information is also used to select a platform, col 12, lines 19-22). Therefore,

it would have been obvious to one of ordinary skill in the art at the time the invention was made to include this feature to the existing system so that the software system will be compatible over various platforms.

4. Claims 6 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nelson (US 2001/0049686) in view of Hinsley (US 5,930,511), and further in view of You (US 6,158,045).

Referring to claims 6 and 20, Nelson fails to explicitly teach having each set of context information including a value for each of a set of processor registers associated with the corresponding software task. However, You teaches having a variety of platforms which may vary in the number of processor registers (platforms, number of processor registers, col 9, line 15-26). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include this feature to the existing system so that the software system will be compatible over various platforms.

Referring to claims 11 and 25, Nelson fails to explicitly teach where the native threads support routine enables the threads interface layer to:

- stop an execution of a particular thread and to clean up a set of structures associated with the particular software task

It would have been obvious to one of ordinary skill in the art at the time the invention was made to stop the execution of a particular thread in order to clean up a set of structures associated with the particular thread because it is more easy to do than during execution.

5. Claims 7-10, 14, 15, 21-24, 28, and 29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nelson (US 2001/0049686) in view of Hinsley (US 5,930,511), and further in view of Gillespie (US 6,269,391).

Referring to claims 7 and 21, Nelson fails to explicitly teach of where the native threads support routine enables the threads interface layer to:

- suspend a particular software task.

However, Gillespie teaches this limitation by disclosing a multiprocessor which suspends execution of a thread (40, col 4, lines 44-46). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the suspend feature of Gillespie to the existing system for the reason of giving threads more control by being able to stop or wait (stop/wait, col 4, line 45).

Referring to claims 8 and 22, Nelson fails to explicitly teach of having a native threads support routine enables the threads interface layer to:

- resume a particular software task

However, Gillespie teaches resuming a thread to initiate or continue execution of the thread (“resume 42 (start), to initiate or continue execution of the thread, col 4, lines 44-49). It would

have been obvious to one of ordinary skill in the art at the time the invention was made to combine the resume feature of Gillespie to the existing system for the reason of giving threads more control by being able to start execution (col 4, line 46).

Referring to claims 9 and 23, Nelson fails to explicitly teach of where the native threads support routine enables the threads interface layer to:

- wait for completion of a particular software task

However, Gillespie teaches that individual threads can have wait times, and thus, it should be known how long of a wait a particular thread would take to be completed (“individual thread’s wait times,” col 8, lines 51-53). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the “wait for completion” feature of Gillespie to the existing system for the reason of giving threads more control by being able to coordinate CPU-specific scheduling data (col 8, lines 43-59).

Referring to claims 10 and 24, Nelson fails to explicitly teach of having a native threads support routine enables the threads interface layer to:

- yield execution to another software task

However, Gillespie teaches using a “yield” to relinquish control (“yield 44 to relinquish control of the processor”, col 4, lines 44-49), and yielding execution to another thread is a type of control. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the suspend feature of Gillespie to the existing system for the

reason of gaining more control of the processor (“yield 44 to relinquish control of the processor”, col 4, lines 44-49.

Referring to claims 14 and 28, Nelson fails to explicitly teach of having a native threads interface layer include a method that enables the threads interface layer to:

- obtain an identifier of a currently executing thread

However, Gilliespie teaches having a CPU assignment identifier (“CPU assignment identifier 134”, col 7, lines 46-55), which serves the purpose of identifying such assignments that could include threads. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the “thread identifier” feature of Gillespie to the existing system so that a particular thread can be identified/tracked during execution.

Referring to claims 15 and 29, Nelson fails to explicitly teach of having a native threads interface layer include a method that enables the threads interface layer to:

- select a thread for execution

However, Gillespie teaches issuing a request (48) for a thread (50) (col 7, lines 42-46) and having a resume or start executable (76) to indicate if a thread is ready to run (col 7, lines 46-47). When the system determines when the thread is ready to run, it “selects” the thread for execution. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the “select a thread for execution” feature of Gillespie to the existing system for the reason of giving threads more control during execution by being able to select between various threads.

6. Claims 12, 13, 26, and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nelson (US 2001/0049686) in view of Hinsley (US 5,930,511), and further in view of Farrell (US 5,630,128).

Referring to claims 12 and 26, Nelson fails to explicitly teach where having a native threads support routine enables the threads interface layer to:

- set a priority of a particular software task

However, Farrell teaches setting a priority of a particular thread (“select each thread’s priority”, col 4, lines 3-18). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the “set a priority of a particular thread” feature of Farrell to the existing system for the reason of giving threads more control by being able to favor certain threads over others.

Referring to claims 13 and 27, Nelson fails to explicitly teach where having a native threads support routine enables the threads interface layer to:

- obtain priority of particular software task

However, Farrell teaches that “the highest priority thread on the run list is executed first” (col 4, lines 8-9) which shows that a priority is distinguished and obtained in a thread. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the “obtain priority of a particular thread” feature of Farrell to the existing system for

the reason of giving threads more control by being able to identify priority so that certain threads can be favored over others.

ARGUMENTS

Applicant argues Nelson does not disclose or suggest a native threads interface layer that includes a routine for adapting parallel execution of software tasks written for a standard threads interface of a virtual machine to an underlying platform. Rather than adapt software tasks of a virtual machine to an underlying platform, Nelson discloses only the threads support of an underlying platform. With respects to limitations regarding “parallel execution,” attention is directed to the rejections of the according claims because new issue has been presented.

Applicant argues that claims 3-6, 11, 17-20 and 25 are not obvious in view of Nelson and Delagi because Delagi merely discloses processor registers rather than a native threads interface layer that includes a routine for adapting parallel execution of software tasks written for a standard threads interface of a virtual machine to an underlying platform. With respects to limitations regarding “parallel execution,” attention is directed to the rejections of the according claims because new issue has been presented.

Applicant argues that Travis discloses a system for invoking application programs on remote platforms rather than a native threads interface layer that includes a routine for adapting parallel execution of software tasks written for a standard threads interface of a virtual machine

to an underlying platform. With respects to limitations regarding “parallel execution,” attention is directed to the rejections of the according claims because new issue has been presented.

Applicant argues that You discloses a software debugging system rather than a native threads interface layer that includes a routine for adapting parallel execution of software tasks written for a standard threads interface of a virtual machine to an underlying platform. With respects to limitations regarding “parallel execution,” attention is directed to the rejections of the according claims because new issue has been presented.

Applicant argues that Gillespie discloses virtual machine thread scheduling in a multi-processor system rather than a native threads interface layer that includes a routine for adapting parallel execution of software tasks written for a standard threads interface of a virtual machine to an underlying platform. With respects to limitations regarding “parallel execution,” attention is directed to the rejections of the according claims because new issue has been presented.

Applicant argues that Farrel discloses a system for scheduling threads rather than a native threads interface layer that includes a routine for adapting parallel execution of software tasks written for a standard threads interface of a virtual machine to an underlying platform. With respects to limitations regarding “parallel execution,” attention is directed to the rejections of the according claims because new issue has been presented.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kenneth Tang whose telephone number is (703) 305-5334. The examiner can normally be reached on 8:30am-6:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Follansbee can be reached on (703)305-8498. The fax phone numbers for the organization where this application or proceeding is assigned are none for regular communications and none for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is none.

kt

January 27, 2003


MASHOBAN BANUZHAI
PRIMARY EXAMINER